REMARKS

The Examiner states that the declaration is defective because it does not include the names "John Buehler" and "David Johnson" and that a new oath or declaration is required. Applicants respectfully traverse the requirement of a new oath or declaration. Pursuant to 37 C.F.R. § 1.48(f)(1), the filing of the declaration acted to correct the earlier identification of inventorship in the application. Applicants believe that the present amendment to the specification at page 1, lines 2-6 further clarifies the correct inventorship. Withdrawal of the requirement of a new oath or declaration is respectfully requested.

The drawings are objected to under 37 C.F.R. § 1.83(a). Applicants have separately submitted proposed changes to Figures 1, 2, and 5 of the application drawings, with proposed changes shown in red, in accordance with 37 C.F.R. § 1.121(d). Applicants believe that the present proposed changes to Figures 1, 2, and 5 of the application drawings overcome the objection.

The drawings are objected to under 37 C.F.R. § 1.84(p)(5). Applicants have submitted proposed changes to Figure 2 of the application drawings, with proposed changes shown in red, in accordance with 37 C.F.R. § 1.121(d). Applicants believe that the present proposed changes to Figure 2 of the application drawings and the present amendments to the specification at page 30, lines 2-9 and page 35, lines 12-17 overcome the objection.

Claims 5 and 12 stand rejected under 35 U.S.C. § 112, second paragraph, and have been amended. The amendment is believed to overcome the rejection, and withdrawal of the rejection is respectfully requested.

Claims 1-4, 6-11, and 13-14 stand rejected under 35 U.S.C. § 102(b) as being anticipated by Ingham (D.B. Ingham, "W3Objects: A Distributed Object-Oriented Web Server," Sixth International World-Wide Web Conference, Santa Clara, California, April, 1997). This rejection is respectfully traversed.

Claims 1 and 8 recite, among other things, interpreting event-driven programs into serial execution code and reinterpreting the serial execution code into second event-driven programs.

Ingham does not disclose a conversion between serial execution code and event-driven programs. At the first bullet on page 2, Ingham states that W3Objects "supports arbitrary allocation of services to processes and processes to machines, in a manner which is completely transparent to users." From this brief statement, it is unclear how the Examiner interprets Ingham to relate to conversion between serial execution code and event-driven programs. Without more information explicitly disclosing a conversion between serial execution code and event-driven programs, Ingham fails to anticipate claims 1 and 8. To anticipate, the prior art reference must disclose, either expressly or inherently, "each and every element as set forth in the claim." *In re Robertson*, 169 F.3d 743, 745 (Fed. Cir. 1999) (quoting *Verdegaal Bros., Inc. v. Union Oil Co.*, 814 F.2d 628, 631 (Fed. Cir. 1987)). Ingham does not expressly disclose a conversion between serial execution code and event-driven programs, nor does it do so inherently:

> To establish inherency, the extrinsic evidence "must make clear that the missing descriptive matter is necessarily present in the thing described in the reference, and that it would be so recognized by persons of ordinary skill. . . . The mere fact that a certain thing may result from a given set of circumstances is not sufficient."

Id. (quoting *Continental Can Co. v. Monsanto Co.*, 948 F.2d 1264, 1268 (Fed. Cir. 1991)). Ingham does not make clear that a conversion between serial execution code and event-driven programs "is necessarily present," so it neither inherently nor expressly discloses such a conversion. Therefore, Ingham does not anticipate claims 1 and 8. Because Ingham does not teach each and every aspect of the claim 1 and 8 invention, it is respectfully submitted that the rejection is improper and should be withdrawn.

Claims 2-4 and 6-7 depend from claim 1 and are allowable for at least the reasons set forth above for claim 1. Claims 9-11 and 13-14 depend from claim 8 and are allowable for at least the reasons set forth above for claim 8.

Claims 5 and 12 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Ingham. This rejection is respectfully traversed. Claim 5 depends from claim 1, and claim 12 depends from claim 8. As noted above, Ingham does not teach a conversion between serial execution code and event-driven programs. Therefore, applicants respectfully submit that claim 5 is allowable for at least the reasons set forth above for claim 1 and further in view of the novel features claimed therein.

Similarly, applicants respectfully submit that claim 12 is allowable for at least the reasons set forth above for claim 8 and further in view of the novel features claimed therein. Therefore, it is respectfully submitted that the rejection is improper and should be withdrawn.

Applicants respectfully submit that the instant application is in condition for allowance. If the Examiner feels, however, that further amendment and/or discussion may be helpful in facilitating prosecution of the case, the Examiner is respectfully requested to telephone the undersigned attorney of record at the telephone number appearing below.

Respectfully Submitted,

Christopher R. Glembocki
Registration No. 38,800

BANNER & WITCOFF, LTD.
1001 G. Street, N.W.
Washington, D.C. 20001-4597
202-508-9100

Date: October 3, 2002

MARKED-UP VERSION OF AMENDMENTS

In the Specification:

The paragraph at page 1, lines 2-6 has been deleted.

The paragraph at page 19, line 29 to page 20, line 21 has been replaced with the following:

--Figure 5 shows composition space 1002 and interaction space 1003 as including a non-volatile memory 1030 in at least one of the two spaces 1002 and 1003. An ASP page [1045] 1025 containing objects is stored in non-volatile memory 1030. The ASP page [1045] 1025 includes objects balloon, cube, and slab. These objects may have associated methods and properties. In response to a request from a browser in the interaction space 1003, the ASP page [1045] 1025 executes and instantiates (step 1031) the object "balloon" with stored property information from non-volatile memory 1030 so as to create balloon$_1$ 1032. Balloon$_1$ 1032 is then transmitted (step 1033) to interaction space 1003 and appears as balloon$_1$ 1034 with persistence information 1035. In this example, the object balloon$_1$ 1034 is transmitted with its own code to interaction space 1003. The persistence information 1035 that accompanies balloon$_1$ 1034 describes the properties of received balloon$_1$ 1034. Here, the properties are (object number =1, color = red, radius = 5). Next, in the interaction space 1003, the browser receives an event 1036 (for example, a request to paint the balloon$_1$ 1034 blue). The balloon$_1$ 1034 has not changed but, having received event 1036, is identified as balloon$_1$ [1035] 1037. The browser transmits (step 1038) the event 1036 to the composition space to handle the event. The information transmitted to the composition space 1002 includes the event 1036 (paint blue) and the persistence information 1035 which identifies the balloon$_1$ [1035] 1037 to the ASP in composition space 1002. Next, the ASP re-instantiates (step 1039) the object balloon$_1$ as 1041 and handles the event 1036 (paint blue) in step 1040. The object balloon$_1$ is transformed into balloon$_2$ 1042. The ASP then transmits balloon$_2$ 1042 (in step 1043) back to the interaction space 1003 with persistence information (object = 1, color = blue, radius = 5) 1045. Finally, the balloon$_2$ is received as balloon$_2$ 1044 and displayed to a user. In this example, it is noted that at least one advantage of storing the persistence information in the interaction space 1003 is that the interaction space is the best place to maintain this information.--

The paragraph at page 30, lines 2-9 has been replaced with the following:

--Figure 9 relates to the execution of a method with no new navigation in the client's browser. Figure 9 shows the receipt of a page in step [401] 501. Next, the end user selects a method with the name of <foo.execute.method ( )> in step 502. In contrast to step 403, the browser keeps the current page (step 503), and grabs the new page (504), which accesses the method *method*. Finally, while the method *method* may be running elsewhere (or it may be running on the client), the client's browser receives the data generated by the method and returns them to the currently running page (505). To the extent that information is displayed by the new method, the method effectively appears to run in the current page.--

The paragraph at page 35, lines 12-17 has been replaced with the following:

--Figure 11 shows an ASP proxy object 704 located on server 701 and an ASP proxy object 705 located on a client 702. When an ASP object is loaded by a server, the property values are retrieved 706 (from wherever they are stored 703) and are placed into the proxy objects properties 704. Also, the proxy object's properties for the client are prepared and stored in ASP object 704 as well. During construction of the page, the proxy objects' properties may be accessed and modified by the server script. If modified, the properties may be sent back to the data storage 703 for future retrieval.--

In the Claims:

The claims have been amended as follows:

5. (Amended) The apparatus of claim [1] 3, said apparatus further comprising:

design-time controls for controlling the generation of said objects when said design-time controls are placed within said first event-driven programs.

12. (Amended) The method of claim [8] 10, further comprising the steps of:

controlling the generation of said objects with controls that operate during a design time when said controls are placed within said first event-driven programs.